L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
_$

Ps
--

YZ

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$

Z$
```

```
                                                                    H 16
**FILE**ID**SYSIMGFIX




  SSSSSSSS  YY      YY    SSSSSSSS    IIIIII    MM      MM    GGGGGGGG  FFFFFFFFFF    IIIIII    XX      XX
  SSSSSSSS  YY      YY    SSSSSSSS    IIIIII    MM      MM    GGGGGGGG  FFFFFFFFFF    IIIIII    XX      XX
SS          YY      YY  SS              II      MMMM  MMMM  GG          FF              II      XX      XX
SS          YY      YY  SS              II      MMMM  MMMM  GG          FF              II      XX      XX
SS            YY  YY    SS              II      MM  MM  MM  GG          FF              II        XX  XX
SS            YY  YY    SS              II      MM  MM  MM  GG          FF              II        XX  XX
  SSSSSS        YY        SSSSSS        II      MM      MM  GG          FFFFFFFF        II          XX
  SSSSSS        YY        SSSSSS        II      MM      MM  GG          FFFFFFFF        II          XX
        SS      YY              SS      II      MM      MM  GG  GGGGGG  FF              II        XX  XX
        SS      YY              SS      II      MM      MM  GG  GGGGGG  FF              II        XX  XX
        SS      YY              SS      II      MM      MM  GG      GG  FF              II      XX      XX      ....
        SS      YY              SS      II      MM      MM  GG      GG  FF              II      XX      XX      ....
SSSSSSSS        YY      SSSSSSSS      IIIIII    MM      MM    GGGGGG    FF            IIIIII    XX      XX      ....
SSSSSSSS        YY      SSSSSSSS      IIIIII    MM      MM    GGGGGG    FF            IIIIII    XX      XX      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .TITLE  SYS$IMGFIX - Address Fixup System Service
0000            .IDENT  'V04-000'
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ;++
0000    ; Facility:
0000    ;
0000    ;       Executive - Image Activator Completion Routines
0000    ;
0000    ; Abstract:
0000    ;
0000    ;       This module contains subroutines used by the image activator
0000    ;       to perform address relocation after images have been activated.
0000    ;
0000    ; Environment:
0000    ;
0000    ;       Most of the code in this module runs in user mode but some routines
0000    ;       may also be called from exec mode.
0000    ;
0000            .SUBTITLE       History
0000    ;
0000    ; Author:
0000    ;
0000    ;       Lawrence J. Kenah
0000    ;
0000    ; Creation Date:
0000    ;
0000    ;       19 March 1984
0000    ;
0000    ; Modified by:
0000    ;
0000    ;       V03-010 LJK0279         Lawrence J. Kenah       8-May-1984
0000    ;               Miscellaneous cleanup. Remove temporary definition of
0000    ;               SHL$B_SHL_SIZE. Put all code into YF$$SYSIMGACT program
0000    ;               section.
```

```
0000    ;
0000    ;       V03-009 LJK0270         Lawrence J. Kenah       31-Mar-1984
0000    ;               Add code to call shareable image initialization routines.
0000    ;
0000    ;       V03-008 LJK0275         Lawrence J. Kenah       25-Mar-1984
0000    ;               The size of SHL elements is variable. It depends on when
0000    ;               the image was linked.
0000    ;
0000    ;       V03-007 LJK0238         Lawrence J. Kenah       26-Jul-1983
0000    ;               Use new concept of image base address instead of first address
0000    ;               into which image is mapped.
0000    ;
0000    ;       V03-006 LJK0218         Lawrence J. Kenah       28-Jun-1983
0000    ;               Minor cleanup.
0000    ;
0000    ;       V03-005 LJK0200         Lawrence J. Kenah       14-Jun-1983
0000    ;               Make changes that support new image activator
0000    ;
0000    ;               Base addresses of shareable images are now located by searching
0000    ;               the ICB list, a much simpler list than the master fixup vector
0000    ;               list. Routine COPY_SHL is no longer needed. All code that
0000    ;               existed to support a previous design for mapping shareable
0000    ;               images permanently into P1 space is also eliminated. Use
0000    ;               IMG$ prefix for global entry point names. Eliminate prefix
0000    ;               from routines that are local.
0000    ;
0000    ;       V03-004 LJK0195         Lawrence J. Kenah       9-Mar-1983
0000    ;               Make so-called recursive activation capable of activating
0000    ;               more than one image without dropping some fixups on the floor.
0000    ;
0000    ;       V03-003 LJK0192         Lawrence J. Kenah       7-Jan-1983
0000    ;               Do poor man's recursive activation to support shareable
0000    ;               images that reference other shareable images not known
0000    ;               to the image header of the executable image.
0000    ;
0000    ;       V03-002 MLJ0099         Martin L. Jack, 20-Oct-1982  19:40
0000    ;               Fix broken BSBWs.
0000    ;
0000    ;       V03-001 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    ;               Added $SSDEF.
0000    ;
0000    ;--
```

```
0000                   .SUBTITLE       Declarations
0000
0000        ; Include Files:
0000
0000                   $IACDEF                         ; Image activator control flags
0000                   $IAFDEF                         ; Offsets into image activator fixup
0000                                                   ;  area within image file
0000                   $ICBDEF                         ; Image control block offsets
0000                   $IMAGCTXDEF                     ; Context of currently executing image
0000                   $IMGDEF                         ; Image activator status codes
0000                   $PSLDEF                         ; PSL field definitions and constants
0000                   $SETPRTDEF                      ; Argument list offsets for $SETPRT system service
0000                   $SHLDEF                         ; Offsets into shareable image list element
```

```
                                0000          .SBTTL  EXE$IMGFIX      Address Relocation Fixup System Service
                                0000     ;+
                                0000     ; Functional Description:
                                0000     ;
                                0000     ;        This procedure is called after an image is activated but before it
                                0000     ;        is called in order to allow address fixups to be performed in user
                                0000     ;        access mode if the caller of the image so wishes. This prevents
                                0000     ;        process destruction os security breaches if the image that has just
                                0000     ;        been activated contains garbage or selected cleverness.
                                0000     ;
                                0000     ; Calling Sequence:
                                0000     ;
                                0000     ;        CALLS   #0,EXE$IMGFIX
                                0000     ;
                                0000     ; Implicit Input:
                                0000     ;
                                0000     ;        Address space of image just activated that contains fixup vectors
                                0000     ;        that describe the address fixups that must be done.
                                0000     ;
                                0000     ; Implicit Output:
                                0000     ;
                                0000     ;        All appropriate locations are relocated to reflect the locations
                                0000     ;        of each shareable image during this activation of the image.
                                0000     ;
                                0000     ; Completion Codes:
                                0000     ;
                                0000     ;        R0 low bit set => successful completion (SS$_NORMAL)
                                0000     ;
                                0000     ;        R0 low bit clear => error occurred
                                0000     ;
                                0000     ;                Actual error status codes are returned by GET_BASE_ADDRESSES and
                                0000     ;                PROCESS_FIXUP_LIST.
                                0000     ;
                                0000     ; Side Effects:
                                0000     ;
                                0000     ;        See the routine headers for the two functional routines for a
                                0000     ;        description of the effects of this procedure.
                                0000     ;-
                                0000
                                0000     ; Put all of this module into a separate pageable program section
                                0000
                            00000000          .PSECT  YF$$SYSIMGACT
                                0000
                                0000     EXE$IMGFIX::
                        003C    0000          .WORD   ^M<R2,R3,R4,R5>         ; Save some registers
                  35      10    0002          BSBB    GET_BASE_ADDRESSES      ; Get base address of each image
               26 50      E9    0004          BLBC    R0,10$                  ; Skip hard part if an error occurred
                00AC      30    0007          BSBW    PROCESS_FIXUP_LIST      ; Do the actual relocation
                  10      E1    000A          BBC     #IMAGCTX$V_SETVECTOR,-
         1B 00000000'GF         000C                  G^IAC$GL_IMAGCTX,10$    ; Any vectors to set?
                                0012          $IMGACT_S       0,0,0,-
                                0012                  IMGCTL=#IAC$M_SETVECTOR ; Let image activator set them
                  11      E1    002D     10$: BBC     #IMAGCTX$V_INITIALIZE,-
         03 00000000'GF         002F                  G^IAC$GL_IMAGCTX,20$    ; Any routines to be called?
                0189      30    0035          BSBW    INISHRIMG               ; Find them and call them
                          04    0038     20$: RET                             ; Return with final status
```

```
                            0039          .SBTTL  GET_BASE_ADDRESSES - Locate Each Shareable Image
                            0039   ;+
                            0039   ; Functional Description:
                            0039   ;
                            0039   ;       This routine is called before the actual fixup operations are performed
                            0039   ;       to determine the base address of each shareable image that has been
                            0039   ;       mapped. If a shareable image in the fixup list has no corresponding
                            0039   ;       entry of the same name in the master ICB list, an error is reported.
                            0039   ;
                            0039   ;       Note that the image activator has filled in the base address for SHL
                            0039   ;       entry 0, the SHL associated with the image itself.
                            0039   ;
                            0039   ; Calling Sequence:
                            0039   ;
                            0039   ;       JSB     GET_BASE_ADDRESSES
                            0039   ;
                            0039   ; Input Parameters:
                            0039   ;
                            0039   ;       none
                            0039   ;
                            0039   ; Implicit Input:
                            0039   ;
                            0039   ;       Listheads for fixup vector list and ICB list
                            0039   ;
                            0039   ; Output Parameters:
                            0039   ;
                            0039   ;       none
                            0039   ;
                            0039   ; Implicit Output:
                            0039   ;
                            0039   ;       All SHL entries in the linked list of fixup vectors have base addresses
                            0039   ;       of their associated shareable images stored in SHL$L_BASEVA.
                            0039   ;
                            0039   ; Completion Codes:
                            0039   ;
                            0039   ;       R0 = SS$_NORMAL
                            0039   ;
                            0039   ;               All base addresses were successfully stored.
                            0039   ;
                            0039   ;       R0 = IMG$_IMAGE_NOT_FOUND
                            0039   ;
                            0039   ;               A shareable image name in a SHL entry had no corresponding
                            0039   ;               ICB. This means that the shareable image was not mapped,
                            0039   ;               which indicates an inconsistency between SHL entries and
                            0039   ;               image section descriptors in the image header of one of the
                            0039   ;               images that was mapped.
                            0039   ;
                            0039   ; Side Effects:
                            0039   ;
                            0039   ;       R0 and R1 are destroyed
                            0039   ;-
                            0039
                            0039   GET_BASE_ADDRESSES:
     55   FFFFFFFC'GF   DE  0039          MOVAL   G^<CTL$GL_FIXUPLNK-IAF$L_FIXUPLNK>,R5 ; Pick up listhead address
                            0040
       55    04 A5      D0  0040   10$:   MOVL    IAF$L_FIXUPLNK(R5),R5    ; Get address of next fixup vector
                27      13  0044          BEQL    30$                      ; Return success if done
```

```
         52    1C A5   D0  0046               MOVL    IAF$L_SHRIMGCNT(R5),R2    ; Count of SHL entries to R2
                  F4   13  004A               BEQL    10$                       ; None here. Get next fixup vector
   53    18 A5    55   C1  004C               ADDL3   R5,IAF$L_SHLSTOFF(R5),R3 ; Address of first SHL entry to R3
         54    10 A3   9A  0051               MOVZBL  SHL$B_SHL_SIZE(R3),R4     ; Get size of each SHL element
                           0055
                           0055       ; By jumping into the middle of the loop, we are in effect skipping over
                           0055       ; entry 0, whose base address was stored by the image activator when the
                           0055       ; image was mapped.
                           0055
                  0E   11  0055               BRB     25$
                           0057
         50    18 A3   9E  0057       20$:    MOVAB   SHL$T_IMGNAM(R3),R0       ; Pass shareable image name in R0
                0015   30  005B               BSBW    IMG$IS_IT_MAPPED          ; Find associated SHL entry in ICB LIST
               11 50   E9  005E               BLBC    R0,40$                    ; Quit if error occurred
         63    5C A1   D0  0061               MOVL    ICB$L_BASE_ADDRESS(R1),SHL$L_BASEVA(R3)
                           0065                                                 ; Store base address
            53    54   C0  0065       25$:    ADDL2   R4,R3                     ; Point to next SHL entry
               EC 52   F5  0068               SOBGTR  R2, 20$                   ;  and do next entry
                           006B
                  D3   11  006B               BRB     10$                       ; Go back and get next fixup vector
                           006D
       50    0000'8F   3C  006D       30$:    MOVZWL  #SS$_NORMAL,R0            ; Indicate success to caller
                       05  0072       40$:    RSB                               ;  and return
```

```
                        0073            .SUBTITLE       IMG$IS_IT_MAPPED - Search ICB List for Shareable Image
                        0073    ;+
                        0073    ; Functional Description:
                        0073    ;
                        0073    ;       This routine searches the shareable image list associated with the
                        0073    ;       executable image to determine whether a shareable image with a given
                        0073    ;       name exists in the list. This routine is used to determine whether a
                        0073    ;       shareable image has already been mapped. Is is also used to relate the
                        0073    ;       relative shareable image list within a shareable image to the shareable
                        0073    ;       image list associated with the executable image.
                        0073    ;
                        0073    ; Calling Sequence:
                        0073    ;
                        0073    ;       JSB     IMG$IS_IT_MAPPED
                        0073    ;
                        0073    ; Input Parameters:
                        0073    ;
                        0073    ;       R0 = address of counted (ASCIC) string of shareable image name
                        0073    ;
                        0073    ; Implicit Input:
                        0073    ;
                        0073    ;       IAC$GL_IMAGE_LIST - Doubly linked list of ICBs describing images
                        0073    ;               that have already been mapped
                        0073    ;
                        0073    ; Output Parameters:
                        0073    ;
                        0073    ;       If successful, R1 contains the address of the image control block
                        0073    ;       that describes the named image.
                        0073    ;
                        0073    ; Completion Codes:
                        0073    ;
                        0073    ;       R0 low bit set indicates success (SS$_NORMAL)
                        0073    ;
                        0073    ;       R0 low bit clear indicates failure (IMG$_IMAGE_NOT_FOUND)
                        0073    ;
                        0073    ;               This status indicates that no match occurred, implying that
                        0073    ;               the shareable image in question has not yet been mapped.
                        0073    ;
                        0073    ;               The severity of this status depends on the caller. This routine
                        0073    ;               is called by the image activator to determine whether an image
                        0073    ;               has been mapped. If the image name is not found, then the image
                        0073    ;               activator maps the image. If this routine returns this status
                        0073    ;               to the fixup code located earlier in this module, that is a
                        0073    ;               fatal error indicating an inconsistence between shareable image
                        0073    ;               lists inside fixup vectors and ISD lists in image headers.
                        0073    ;-
                        0073
                        0073    IMG$IS_IT_MAPPED::
          00FC 8F   BB  0073            POSHR   #^M<R2,R3,R4,R5,R6,R7>  ; Save some registers
            54 80   9A  0077            MOVZBL  (R0)+,R4                ; Save character count in R4
            55 50   D0  007A            MOVL    R0,R5                   ; Save string address in R5
                        007D
                        007D            ASSUME  ICB$L_FLINK EQ 0
                        007D
    57   00000000'GF   DE  007D            MOVAL   G^IAC$GL_IMAGE_LIST,R7  ; Get address of ICB listhead
            56 57   D0  0084            MOVL    R7,R6                   ; Copy it to a working register
                        0087
```

```
                  56    66    D0  0087    310 10$:    MOVL    ICB$L_FLINK(R6),R6        ; Get address of next ICB
                  57    56    D1  008A    311         CMPL    R6,R7                     ; Check for end of list
                        1C    13  008D    312         BEQL    30$                       ; Equality indicates no more ICBs
                                  008F    313
               14 A6    54    91  008F    314         CMPB    R4,ICB$T_IMAGE_NAME(R6)   ; Do string sizes agree?
                        F2    12  0093    315         BNEQ    10$                       ; No, go get next ICB
          15 A6   65    54    29  0095    316         CMPC3   R4,(R5),ICB$T_IMAGE_NAME+1(R6)   ; Check strings for equality
                        EB    12  009A    317         BNEQ    10$                       ; Go get next ICB if no match
                  51    56    D0  009C    318         MOVL    R6,R1                     ; Store ICB address
                                  009F    319
          50   00000000'8F    D0  009F    320         MOVL    #SS$_NORMAL,R0            ; Indicate success to caller
                  00FC 8F     BA  00A6    321 20$:    POPR    #^M<R2,R3,R4,R5,R6,R7>    ; Restore registers
                        05        00AA    322         RSB                               ;  and return
                                  00AB    323
                                  00AB    324 ; If we loop through the entire ICB list without matching the image name, then
                                  00AB    325 ; the shareable image has not yet been mapped. Indicate that to caller.
                                  00AB    326
          50   084D8962 8F    D0  00AB    327 30$:    MOVL    #IMG$_IMAGE_NOT_FOUND,R0
                        51    D4  00B2    328         CLRL    R1
                        F0    11  00B4    329         BRB     20$
```

```
                          0086          .SBTTL  PROCESS_FIXUP_LIST - Perform Post-Activation Fixups
                          0086     ;+
                          0086     ; Functional Description:
                          0086     ;
                          0086     ;       This routine processes a linked list of fixup vectors and performs
                          0086     ;       the specific fixup operations listed in each vector. There are three
                          0086     ;       forms of fixup.
                          0086     ;
                          0086     ;               o  Each exit vector has the base address of the shareable
                          0086     ;                  image added to each entry.
                          0086     ;
                          0086     ;               o  Each .ADDRESS directive has the base address of the
                          0086     ;                  appropriate shareable image added to it.
                          0086     ;
                          0086     ;               o  The protection of each fixup vector is changed to prevent
                          0086     ;                  its being written when the image executes.
                          0086     ;
                          0086     ; Calling Sequence:
                          0086     ;
                          0086     ;       JSB     PROCESS_FIXUP_LIST
                          0086     ;
                          0086     ; Input Parameters:
                          0086     ;
                          0086     ;       none
                          0086     ;
                          0086     ; Implicit Input:
                          0086     ;
                          0086     ;       CTL$GL_FIXUPLNK         Listhead of linked list of fixup vectors for
                          0086     ;                               a set of shareable images
                          0086     ;
                          0086     ; Output Parameters:
                          0086     ;
                          0086     ;       none
                          0086     ;
                          0086     ; Implicit Output:
                          0086     ;
                          0086     ;       Elements in fixup vector G-hat offset area have base address
                          0086     ;       of appropriate shareable image added to them.
                          0086     ;
                          0086     ;       .ADDRESS directives throughout the address space have base
                          0086     ;       addresses added in.
                          0086     ;
                          0086     ;       Pages that should eventually be read-only but were set to
                          0086     ;       writable while the image activator works are set back to read-only.
                          0086     ;
                          0086     ; Completion Codes:
                          0086     ;
                          0086     ;       none
                          0086     ;
                          0086     ; Side Effects:
                          0086     ;
                          0086     ;       CTL$GL_FIXUPLNK cleared after fixups are completed.
                          0086     ;
                          0086     ;--
                          0086
                          0086     PROCESS_FIXUP_LIST:
   55   FFFFFFFC'GF   DE  0086          MOVAL   G^<CTL$GL_FIXUPLNK-IAF$L_FIXUPLNK>,R5 ; Pick up listhead address
```

```
                     00BD   388
55    04 A5   D0     00BD   389 10$:     MOVL     IAF$L_FIXUPLNK(R5),R5     ; Get address of next fixup vector
         2E   13     00C1   390          BEQL     40$                       ; Quit if no more to process
         51   D4     00C3   391          CLRL     R1                        ; Need base address of this shareable
       0042   30     00C5   392          BSBW     SHIMG_BASVA               ;  image (with index 0)
   53    51   D0     00C8   393          MOVL     R1,R3                     ; Load correct input register
54    0C A5   D0     00CB   394          MOVL     IAF$L_G_FIXOFF(R5),R4     ; Get offset to G-hat fixup data
         06   13     00CF   395          BEQL     20$                       ; Skip this step if none
   54    55   C0     00D1   396          ADDL2    R5,R4                     ; Make an address
       001B   30     00D4   397          BSBW     FIXUP_G_HAT               ; Go do the actual work
54    10 A5   D0     00D7   398 20$:     MOVL     IAF$L_DOTADROFF(R5),R4    ; Get offset to .ADDRESS fixup data
         06   13     00DB   399          BEQL     30$                       ; Skip this step if none
   54    55   C0     00DD   400          ADDL2    R5,R4                     ; Make an address
       0039   30     00E0   401          BSBW     FIXUP_ADDRESS             ; Fixup all .ADDRESS data
54    14 A5   D0     00E3   402 30$:     MOVL     IAF$L_CHGPRTOFF(R5),R4    ; Get offset to page protection data
         D4   13     00E7   403          BEQL     10$                       ; Skip this step if none
   54    55   C0     00E9   404          ADDL2    R5,R4                     ; Make an address
       0049   30     00EC   405          BSBW     FIXUP_PROT                ; Change page protection
         CC   11     00EF   406          BRB      10$                       ; All done with this fixup vector
                     00F1   407                                            ;  See if there are any more
              05     00F1   408 40$:     RSB                                ; Return to caller
```

```
                        00F2          .SBTTL  FIXUP_G_HAT     Fixup G-hat exit vector
                        00F2     ;+
                        00F2     ; Functional Description:
                        00F2     ;
                        00F2     ;       This routine performs the G-hat fixup for a specific exit vector.
                        00F2     ;       specifically, the base address of the appropriate shareable image
                        00F2     ;       is added to each entry in the exit vector.
                        00F2     ;
                        00F2     ; Calling Sequence:
                        00F2     ;
                        00F2     ;       BSBW    FIXUP_G_HAT
                        00F2     ;
                        00F2     ; Input Parameters:
                        00F2     ;
                        00F2     ;       R4 = Address of G-hat fixup area within fixup vector
                        00F2     ;
                        00F2     ; Implicit Input:
                        00F2     ;
                        00F2     ;       Contents of G-hat fixup area
                        00F2     ;
                        00F2     ; Output Parameters:
                        00F2     ;
                        00F2     ;       none
                        00F2     ;
                        00F2     ; Implicit Output:
                        00F2     ;
                        00F2     ;       Elements in fixup vector G-hat offset area have base address
                        00F2     ;         of appropriate shareable image added to them.
                        00F2     ;
                        00F2     ; Completion Codes:
                        00F2     ;
                        00F2     ;       none
                        00F2     ;
                        00F2     ; Side Effects:
                        00F2     ;
                        00F2     ;       R0, R1, and R2 are destroyed
                        00F2     ;-
                        00F2
                        00F2     FIXUP_G_HAT:
         52     84  D0  00F2            MOVL    (R4)+,R2                ; R2 contains a count of fixups
                0D  13  00F5            BEQL    20$                     ; A zero indicates the end of the G-hat data
         51     84  D0  00F7            MOVL    (R4)+,R1                ; Store shareable image number in R1
                0E  10  00FA            BSBB    SHIMG_BASVA             ;  and then load R1 with base address
                        00FC                                            ;  of next shareable image.
         84     51  C0  00FC     10$:   ADDL2   R1,(R4)+                ; Bias next exit vector entry
             FA 52  F5  00FF            SOBGTR  R2,10$                  ; Do next entry
                EE  11  0102            BRB     FIXUP_G_HAT             ; Now do next shareable image
                        0104
   50    0000'8F    3C  0104     20$:   MOVZWL  #SS$_NORMAL,R0          ; Indicate success
                    05  0109            RSB                             ; Return
```

```
                                  010A   461          .SBTTL  SHIMG_BASVA     Convert a shareable image index to an address
                                  010A   462 ;+
                                  010A   463 ; Functional Description:
                                  010A   464 ;
                                  010A   465 ;        This routine converts a relative shareable image number into the
                                  010A   466 ;        absolute base address at which that shareable image is mapped. It
                                  010A   467 ;        assumes that the base address of each shareable image has already
                                  010A   468 ;        been stored in its associated SHL entry.
                                  010A   469 ;
                                  010A   470 ; Calling Sequence:
                                  010A   471 ;
                                  010A   472 ;        BSBW    SHIMG_BASVA
                                  010A   473 ;
                                  010A   474 ; Input Parameters:
                                  010A   475 ;
                                  010A   476 ;        R1 = Relative number of shareable image
                                  010A   477 ;        R5 = Base address of fixup vector
                                  010A   478 ;
                                  010A   479 ; Implicit Input:
                                  010A   480 ;
                                  010A   481 ;        Contents of SHL$L_BASEVA for shareable image indexed by R1.
                                  010A   482 ;
                                  010A   483 ; Output Parameters:
                                  010A   484 ;
                                  010A   485 ;        R1 = Base address of shareable image indicated by input parameter
                                  010A   486 ;
                                  010A   487 ; Side Effects:
                                  010A   488 ;
                                  010A   489 ;        R0 is destroyed
                                  010A   490 ;-
                                  010A   491
                                  010A   492 SHIMG_BASVA:
        50    18 A5    55    C1   010A   493          ADDL3   R5,IAF$L_SHLSTOFF(R5),R0 ; Base address of shareable image list
             7E    10 A0    9A   010F   494          MOVZBL  SHL$B_SHL_SIZE(R0),-(SP) ; Get size of each SHL element
     50    50    51    8E    7A   0113   495          EMUL    (SP)+,R1,R0,R0           ; R0 points to correct SHL entry
                   51    60    D0   0118   496          MOVL    SHL$L_BASEVA(R0),R1      ; Store associated base address
                               05   011B   497          RSB                              ;  and return
```

```
                            011C               .SBTTL  FIXUP_ADDRESS   Fixup .ADDRESS entries throughout the image
                            011C       ;+
                            011C       ; Functional Description:
                            011C       ;
                            011C       ;       This routine performs the .ADDRESS fixup for a specific exit vector.
                            011C       ;       Specifically, the base address of the appropriate shareable image
                            011C       ;       is added to each .ADDRESS entry in this shareable image.
                            011C       ;
                            011C       ; Calling Sequence:
                            011C       ;
                            011C       ;       BSBW    FIXUP_ADDRESS
                            011C       ;
                            011C       ; Input Parameters:
                            011C       ;
                            011C       ;       R3 = Base address of shareable image whose .ADDRESS directives
                            011C       ;               are being fixed
                            011C       ;       R4 = Address of .ADDRESS fixup area within fixup vector
                            011C       ;
                            011C       ; Implicit Input:
                            011C       ;
                            011C       ;       Contents of .ADDRESS fixup area
                            011C       ;
                            011C       ; Implicit Output:
                            011C       ;
                            011C       ;       .ADDRESS directives within this shareable image have the base addresses
                            011C       ;       of the appropriate shareable images added to them.
                            011C       ;-
                            011C
                            011C       FIXUP_ADDRESS:
            52    84    D0  011C               MOVL    (R4)+,R2                ; R2 contains a count of fixups
                  11    13  011F               BEQL    20$                     ; A zero indicates the end of the G-hat data
            51    84    D0  0121               MOVL    (R4)+,R1                ; Store shareable image number in R1
                  E4    10  0124               BSBB    SHIMG_BASVA             ;  and then load R1 with base address
                            0126                                               ;  of next shareable image.
      50    84    53    C1  0126       10$:    ADDL3   R3,(R4)+,R0             ; Get address of .ADDRESS directive
            60    51    C0  012A               ADDL2   R1,(R0)                 ; Bias by base address of shareable image
               F6 52    F5  012D               SOBGTR  R2,10$                  ; Do next entry
                  EA    11  0130               BRB     FIXUP_ADDRESS           ; Now do next shareable image
                            0132
      50    0000'8F     3C  0132       20$:    MOVZWL  #SS$_NORMAL,R0          ; Indicate success
                        05  0137               RSB                             ; Return
```

```
                            0138                  .SBTTL  FIXUP_PROT      Alter page protection to read only
                            0138         ;+
                            0138         ; Functional Description:
                            0138         ;
                            0138         ;        This routine alters the page protection of various sections within
                            0138         ;        the image to read only. These pages were initially writable so the
                            0138         ;        image activator could fixup all of the relative references. The pages
                            0138         ;        cannot be writable while the image is executing.
                            0138         ;
                            0138         ; Calling Sequence:
                            0138         ;
                            0138         ;        BSBW    FIXUP_PROT
                            0138         ;
                            0138         ; Input Parameters:
                            0138         ;
                            0138         ;        R3 = Base address of image whose pages' protection is being altered
                            0138         ;        R4 = Address of protection data within fixup vector
                            0138         ;
                            0138         ; Implicit Input:
                            0138         ;
                            0138         ;        Contents of protection data in fixup vector
                            0138         ;
                            0138         ; Implicit Output:
                            0138         ;
                            0138         ;        Pages in address ranges specified in fixup vector have their protections
                            0138         ;        changed to the protections also specified in that data area. The
                            0138         ;        protection is usually no write access for any access mode.
                            0138         ;
                            0138         ; Side Effects:
                            0138         ;
                            0138         ;        R0, R1, and R2 are destroyed
                            0138         ;-
                            0138
                            0138         FIXUP_PROT:
               56      DD   0138                  PUSHL   R6                      ; Need one more register here
        5E     14      C2   013A                  SUBL2   #<4*SETPRT$_NARGS>,SP   ; Set up space for argument list
               05      DD   013D                  PUSHL   #SETPRT$_NARGS          ; Push argument count
        56     5E      D0   013F                  MOVL    SP,R6                   ; Use R6 as argument pointer
               7E      7C   0142                  CLRQ    -(SP)                   ; Initialize input address array
     04 A6     5E      D0   0144                  MOVL    SP,SETPRT$_INADR(R6)    ; Put its address into argument list
        08 A6          D4   0148                  CLRL    SETPRT$_RETADR(R6)      ; Not interested in this argument
     0C A6     01      D0   014B                  MOVL    #PSL$C_EXEC,SETPRT$_ACMODE(R6) ; The image activator owns these page
        14 A6          D4   014F                  CLRL    SETPRT$_PRVPRT(R6)      ; Not interested in this either
  50    0000'8F        3C   0152                  MOVZWL  #SS$_NORMAL,R0          ; Establish initial status
        52     84      D0   0157                  MOVL    (R4)+,R2                ; Get count of number of protection changes
               20      13   015A                  BEQL    20$                     ; Do not even start if nothing here
  6E    84     53      C1   015C         10$:     ADDL3   R3,(R4)+,(SP)           ; Get starting address
        51     84      3C   0160                  MOVZWL  (R4)+,R1                ; Ending address must be calculated
  51    51     09      78   0163                  ASHL    #9,R1,R1                ;  ... from page count in image section
               51      D7   0167                  DECL    R1                      ; Make byte count an inclusive count
04 AE   6E     51      C1   0169                  ADDL3   R1,(SP),4(SP)           ; Put ending address in second longword
     10 A6     84      3C   016E                  MOVZWL  (R4)+,SETPRT$_PROT(R6)  ; Get new protection from fixup vector
00000000'GF    66      FA   0172                  CALLG   (R6),G^SYS$SETPRT       ; Call the system service
                            0179                                                  ; Ignore errors
           E0 52       F5   0179                  SOBGTR  R2,10$                  ; Go get next image section
                            017C
        5E     20      C0   017C         20$:     ADDL2   #<8+4+<4*SETPRT$_NARGS>>,SP ;Reset stack pointer,
                       56 8ED0  017F    598          POPL    R6                              ;  restore that extra register,
                            05  0182    599          RSB                                     ;  and return
```

```
                             0183           .SBTTL  IMG$PRVSHRIMG   Fixup Routine for Privileged Shareable Images
                             0183      ;+
                             0183      ; Functional Description:
                             0183      ;
                             0183      ;        This routine checks that a privileged shareable image has no
                             0183      ;        outbound calls. For images passing this test, remaining
                             0183      ;        .ADDRESS fixups are performed.
                             0183      ;
                             0183      ; Calling Sequence:
                             0183      ;
                             0183      ;        BSBW    IMG$PRVSHRIMG
                             0183      ;
                             0183      ; Input Parameters:
                             0183      ;
                             0183      ;        R0      Address of fixup vector
                             0183      ;        R1      Base address of privileged shareable image currently
                             0183      ;                being mapped
                             0183      ;
                             0183      ; Implicit Output:
                             0183      ;
                             0183      ;        If the fixup vector indicates no outbound calls, the base address
                             0183      ;        of the privileged shareable image is stored in the fixup vector
                             0183      ;        and the .ADDRESS fixups are performed.
                             0183      ;
                             0183      ; Side Effects:
                             0183      ;
                             0183      ;        R0 and R1 are destroyed
                             0183      ;
                             0183      ; Completion Codes:
                             0183      ;
                             0183      ;        SS$_NORMAL      Fixups were completed for privileged shareable image
                             0183      ;
                             0183      ;        SS$_NOSHRIMG    Shareable image has outbound calls
                             0183      ;-
                             0183
                             0183      IMG$PRVSHRIMG::
                  3C    BB   0183              PUSHR   #^M<R2,R3,R4,R5>         ; Save some registers
             55   50    D0   0185              MOVL    R0,R5                    ; Store fixup vector address in R5
  50    1C A5     01    C3   0188              SUBL3   #1,IAF$L_SHRIMGCNT(R5),R0      ; Is shareable imagc count 1?
                  2B    12   018D              BNEQ    30$                      ; If not, report error
          0C A5   D5         018F              TSTL    IAF$L_G_FIXOFF(R5)       ; Also report error if G^ fixup data
                  26    12   0192              BNEQ    30$
             53   51    D0   0194              MOVL    R1,R3                    ; Store base address of image in R3
  50    18 A5     55    C1   0197              ADDL3   R5,IAF$L_SHLSTOFF(R5),R0       ; Also store base address in
             60   51    D0   019C              MOVL    R1,SHL$L_BASEVA(R0)      ;  SHL entry for SHIMG_BASVA
       54    10 A5      D0   019F              MOVL    IAF$L_DOTADROFF(R5),R4   ; Any .ADDRESS fixups?
                  06    13   01A3              BEQL    10$                      ; Branch if none
       54         55    C0   01A5              ADDL2   R5,R4                    ; Convert R4 offset to address
                FF71    30   01A8              BSBW    FIXUP_ADDRESS            ; Fixup all .ADDRESS data
       54    14 A5      D0   01AB      10$:    MOVL    IAF$L_CHGPRTOFF(R5),R4   ; Get offset to protection data
                  06    13   01AF              BEQL    20$                      ; All done if none
       54         55    C0   01B1              ADDL2   R5,R4                    ; Make R4 an address
                FF81    30   01B4              BSBW    FIXUP_PROT               ; Change page protection
                  3C    BA   01B7      20$:    POPR    #^M<R2,R3,R4,R5>         ; Restore registers
                  05         01B9              RSB                              ;  and return
                             01BA
    50    0000'8F       3C   01BA      30$:    MOVZWL  #SS$_NOSHRIMG,R0         ; No outbound calls allowed
```

```
                      F6   11  01BF   658          BRB     20$                     ; Return error status
```

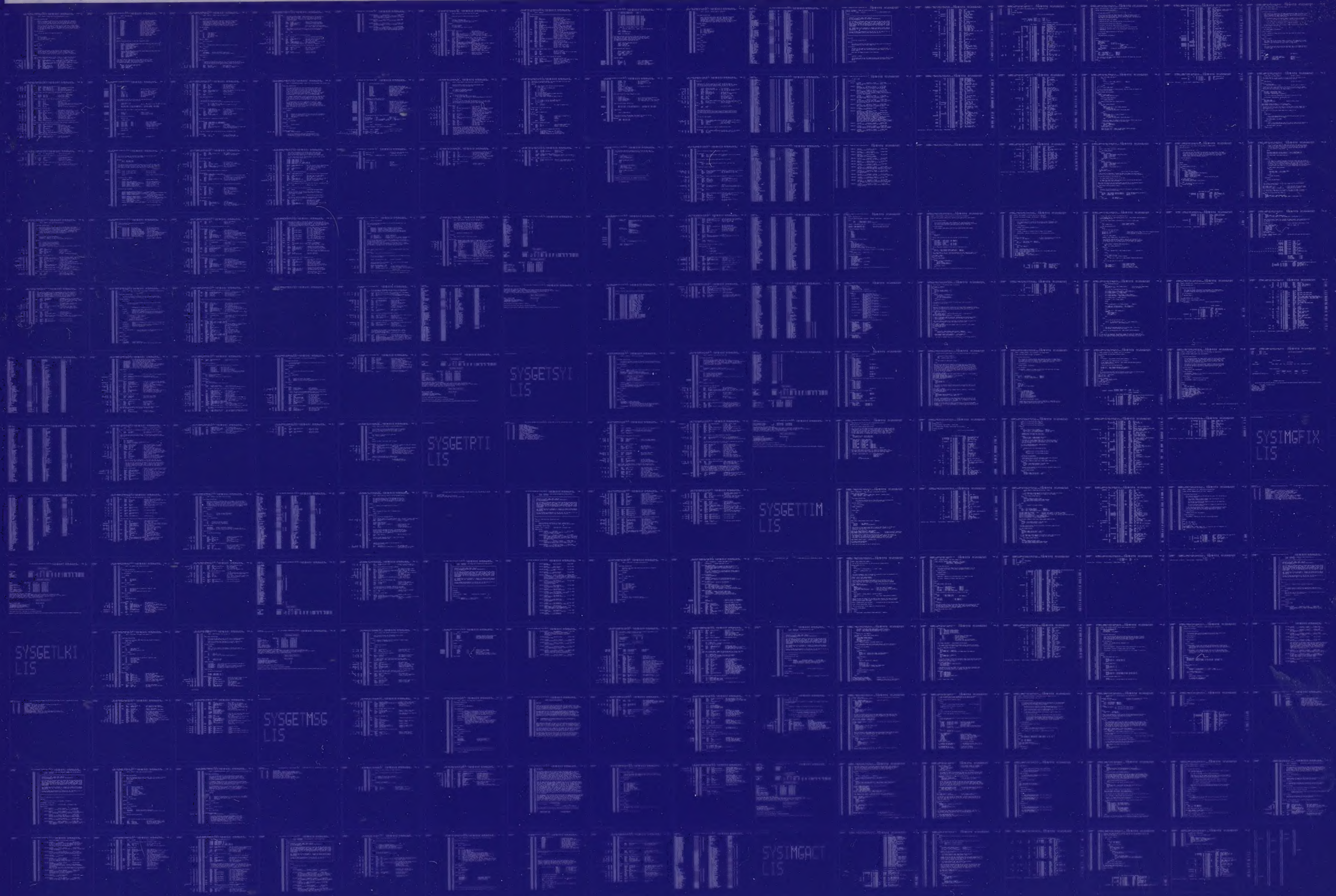
```
                          01C1            .SBTTL  INISHRIMG - Look for and Call Shareable Image Initialization Code
                          01C1      ;+
                          01C1      ; Functional Description:
                          01C1      ;
                          01C1      ;       This routine searches the shareable image list for images that have
                          01C1      ;       included initialization code.
                          01C1      ;
                          01C1      ; Calling Sequence:
                          01C1      ;
                          01C1      ;       BSBW    INISHRIMG
                          01C1      ;
                          01C1      ; Input Parameters:
                          01C1      ;
                          01C1      ;       none
                          01C1      ;
                          01C1      ; Implicit Input:
                          01C1      ;
                          01C1      ;       IAC$GL_IMAGE_LIST - List of ICBs describing shareable images that
                          01C1      ;               are currently mapped.
                          01C1      ;       IAC$GL_FIRST_ICB - Address of ICB representing main image in the
                          01C1      ;               most recent image activation.
                          01C1      ;
                          01C1      ; Implicit Output:
                          01C1      ;
                          01C1      ;       If there are any images with ICBs containing shareable image
                          01C1      ;       initialization code, these procedures are called at their entry
                          01C1      ;       points. Note that the ICB list is traversed backwards.
                          01C1      ;
                          01C1      ; Side Effects:
                          01C1      ;
                          01C1      ;       R0 and R1 are destroyed
                          01C1      ;
                          01C1      ; Completion Codes:
                          01C1      ;
                          01C1      ;       none
                          01C1      ;-
                          01C1
                          01C1      INISHRIMG:
         7E   52     7D   01C1              MOVQ    R2,-(SP)                ; Save some registers
52  00000000'GF      DE   01C4              MOVAL   G^IAC$GL_IMAGE_LIST,R2  ; Get the listhead address
53  00000000'GF      D0   01CB              MOVL    G^IAC$GL_FIRST_ICB,R3   ; This is the stopper
                          01D2
      52   04 A2     D0   01D2      10$:    MOVL    ICB$L_BLINK(R2),R2      ; Get the next ICB
              05     E1   01D6              BBC     #ICB$V_INITIALIZE,-     ; Does this image need to be called?
         09 10 A2         01D8                      ICB$L_FLAGS(R2),20$     ;  Branch if no initialization routine
            60 A2    C1   01DB              ADDL3   ICB$L_INITIALIZE(R2),-  ; Form the address of the entry point
      51    5C A2         01DE                      ICB$L_BASE_ADDRESS(R2),R1
         61   00     FB   01E1              CALLS   #0,(R1)                 ; Call the routine
         53   52     D1   01E4      20$:    CMPL    R2,R3                   ; Is this the end of the line?
              E9     12   01E7              BNEQ    10$                     ; Back to the top is there's more
         52   8E     7D   01E9              MOVQ    (SP)+,R2                ; Restore R2 and R3
                     05   01EC              RSB                             ; All done. Return to caller.
                          01ED
                          01ED              .END
```

```
$$ARGS                            = 00000005
$$T1                              = 00000000
CTL$GL_FIXUPLNK                     ********   X    02
EXE$IMGFIX                          00000000 RG     02
FIXUP_ADDRESS                       0000011C R      02
FIXUP_G_HAT                         000000F2 R      02
FIXUP_PROT                          00000138 R      02
GET_BASE_ADDRESSES                  00000039 R      02
IAC$GL_FIRST_ICB                    ********   X    02
IAC$GL_IMAGCTX                      ********   X    02
IAC$GL_IMAGE_LIST                   ********   X    02
IAC$M_SETVECTOR                   = 00200000
IAF$L_CHGPRTOFF                   = 00000014
IAF$L_DOTADROFF                   = 00000010
IAF$L_FIXUPLNK                    = 00000004
IAF$L_G_FIXOFF                    = 0000000C
IAF$L_SHLSTOFF                    = 00000018
IAF$L_SHRIMGCNT                   = 0000001C
ICB$L_BASE_ADDRESS                = 0000005C
ICB$L_BLINK                       = 00000004
ICB$L_FLAGS                       = 00000010
ICB$L_FLINK                       = 00000000
ICB$L_INITIALIZE                  = 00000060
ICB$T_IMAGE_NAME                  = 00000014
ICB$V_INITIALIZE                  = 00000005
IMAGCTX$V_INITIALIZE              = 00000011
IMAGCTX$V_SETVECTOR               = 00000010
IMG$IS_IT_MAPPED                    00000073 RG     02
IMG$PRVSHRIMG                       00000183 RG     02
IMG$_IMAGE_NOT_FOUND              = 084D8962
INISHRIMG                           000001C1 R      02
PROCESS_FIXUP_LIST                  000000B6 R      02
PSL$C_EXEC                        = 00000001
SETPRT$_ACMODE                    = 0000000C
SETPRT$_INADR                     = 00000004
SETPRT$_NARGS                     = 00000005
SETPRT$_PROT                      = 00000010
SETPRT$_PRVPRT                    = 00000014
SETPRT$_RETADR                    = 00000008
SHIMG_BASVA                         0000010A R      02
SHL$B_SHL_SIZE                    = 00000010
SHL$L_BASEVA                      = 00000000
SHL$T_IMGNAM                      = 00000018
SS$_NORMAL                          ********   X    02
SS$_NOSHRIMG                        ********   X    02
SYS$IMGACT                          ********  GX    02
SYS$SETPRT                          ********   X    02

                                                 +----------------+
                                                 ! Psect synopsis !
                                                 +----------------+

PSECT name                          Allocation          PSECT No.  Attributes
----------                          ----------          ---------  ----------
.  ABS  .                           00000000  (    0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                               00000000  (    0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
YF$$SYSIMGACT                       000001ED  (  493.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
```

```
                                         +--------------------------+
                                         ! Performance in         s !
                                         +--------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        34    00:00:00.07    00:00:00.40
Command processing                   136    00:00:00.73    00:00:03.65
Pass 1                               192    00:00:04.03    00:00:10.67
Symbol table sort                      0    00:00:00.26    00:00:00.34
Pass 2                               133    00:00:01.52    00:00:03.66
Symbol table output                    7    00:00:00.05    00:00:00.05
Psect synopsis output                  2    00:00:00.03    00:00:00.04
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                 506    00:00:06.69    00:00:18.82

The working set limit was 1500 pages.
21961 bytes (43 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 218 non-local and 25 local symbols.
713 source lines were read in Pass 1, producing 14 object records in Pass 2.
20 pages of virtual memory were used to define 19 macros.

                                        +----------------------------+
                                        ! Macro library statistics !
                                        +----------------------------+

Macro library name                      Macros defined
------------------                      --------------
_$255$DUA28:[SYS.OBJ]IMGACT.MLB;1                    3
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                       2
_$255$DUA28:[SYSLIB]STARLET.MLB;2                   11
TOTALS (all libraries)                              16

329 GETS were required to define 16 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSIMGFIX/OBJ=OBJ$:SYSIMGFIX MSRC$:SYSIMGFIX/UPDATE=(ENH$:SYSIMGFIX)+EXECML$/LIB+LIB$:IMGACT/LIB
```

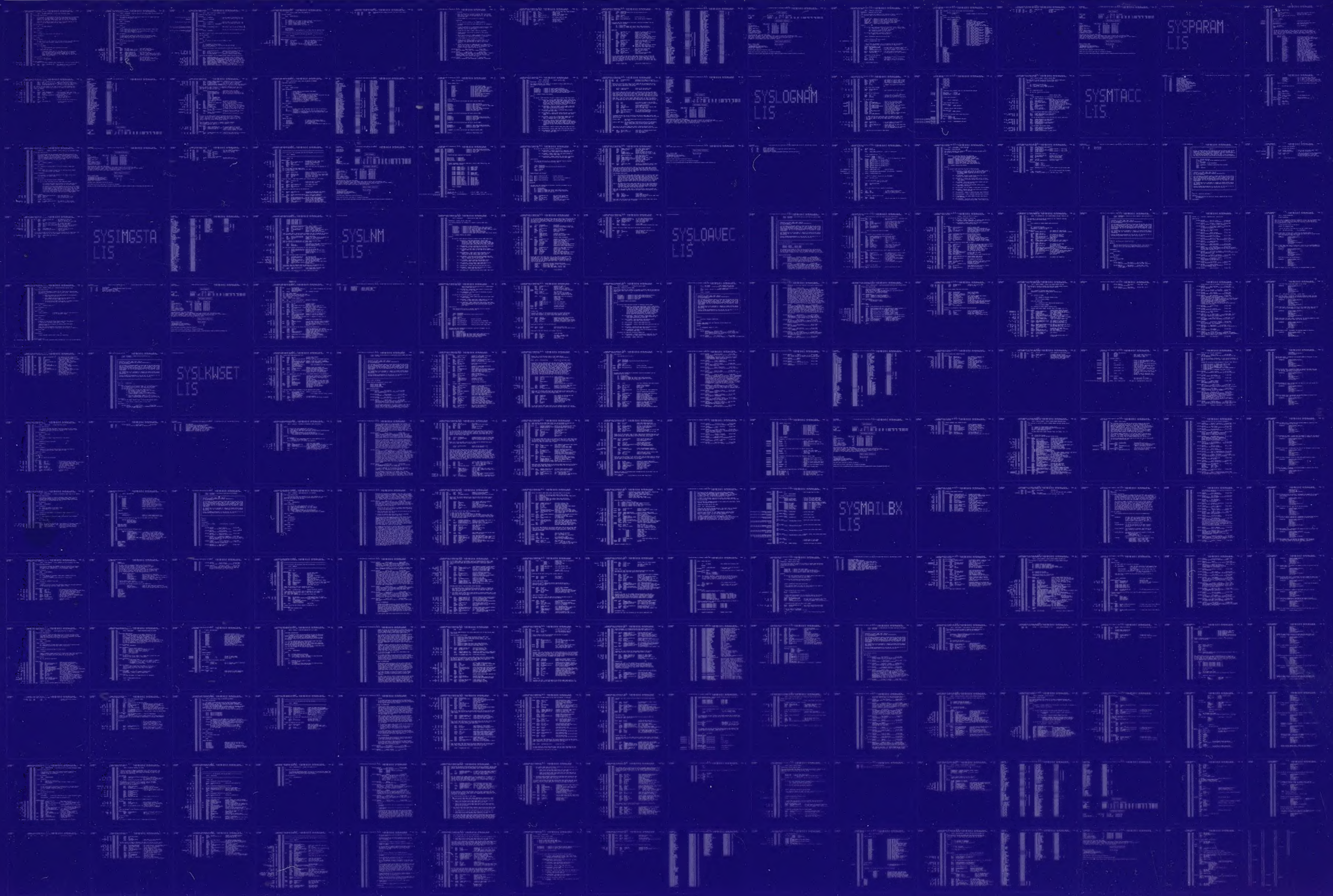
0385 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0386 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



SYSPARAM
LIS

SYSLOGNAM
LIS

SYSMTACC
LIS

SYSIMGSTA
LIS

SYSLNM
LIS

SYSLOAVEC
LIS

SYSLKWSET
LIS

SYSMAILBX
LIS